View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Elsevier - Publisher Connector

ELSEVIER

Theoretical Computer Science 264 (2001) 139–153

Theoretical Computer Science

www.elsevier.com/locate/tcs

Counting by quantum eigenvalue estimation

Michele Mosca *

Mathematical Institute, University of Oxford, 24-29 St. Giles', Oxford OX1 3LB, UK
Clarendon Laboratory, Centre for Quantum Computation, Parks Road, Oxford OX1 3PU, UK

Accepted April 2000

Abstract

For every "computation" there corresponds the physical task of manipulating a starting state into an output state with a desired property. As the classical theory of physics has been replaced by quantum physics, it is interesting to consider the capabilities of a computer that can exploit the distinctive quantum features of nature. The extra capabilities seem enormous. For example, with only an expected $O(\sqrt{N})$ evaluations of a function $f:\{0,1,\ldots,N-1\}\rightarrow\{0,1\}$, we can find a solution to $f(x)=1$ provided one exists. Another example is the ability to find efficiently the order of an element g in a group by using a quantum computer to estimate a random eigenvalue of the unitary operator that multiplies by g in the group. By using this eigenvalue estimation algorithm to estimate an eigenvalue of the unitary operator used in quantum searching we can approximately count the number of solutions to $f(x)=1$. This paper describes this eigenvector approach to quantum counting and related algorithms. Crown Copyright © 2001 Published by Elsevier Science B.V. All rights reserved.

1. Introduction

In [14], Feynman notes that it is unlikely that a classical computer can efficiently simulate the evolution of a quantum system. He thus speculates that a "quantum" computer built to exploit these quantum properties would be much more powerful than a classical computer. Deutsch [12] went on to define the quantum Turing machine and quantum circuits. Evidence that quantum computers are more powerful than classical computers appears in [12, 13, 9, 24]. Building upon the idea of Simon [24], Shor [23] showed how we can use a quantum computer to find the order of an element g from the multiplicative group of integers modulo N for some composite integer N with polylog(N) elementary operations. Shor combines this quantum algorithm with the classical difference of squares factoring technique (see Section 3.2.5 of [21]) to

* Correspondence address: Department of Combinatorics & Optimization, University of Waterloo, Waterloo, ON, Canada, N2L 3G1.

E-mail address: mmosca@cacr.math.uwaterloo.ca (M. Mosca).

0304-3975/01/$ - see front matter Crown Copyright © 2001 Published by Elsevier Science B.V. All rights reserved.
PII: S0304-3975(00)00217-6

produce a quantum factoring algorithm. Consequently, the power of quantum computers became much more tangible. No classical algorithm is known for solving this problem in polynomial time and many public key cryptosystems in use today rely on the computational intractability of factoring (see Chapter 8 of [21]). This order-finding algorithm can be viewed as an estimation of a random eigenvalue of the unitary operator that multiplies by g. This view unifies the approaches of Shor and Kitaev [18] as demonstrated in [10]. The order-finding algorithm is one example in a larger class of algorithms known as *Abelian hidden subgroup* algorithms (see [20] for a survey). They can all be viewed as an estimation of an eigenvalue or a set of eigenvalues of some unitary operator or operators. Implementations can focus on the task of estimating these eigenvalues efficiently.

The other major family of quantum algorithms known are based on Grover's [15] algorithm for quantum searching. These algorithms can be summarised as quantum amplitude amplification [5, 6, 16, 8], quantum amplitude estimation [8, 19], and special cases thereof. The main contributions of this paper are derived by considering the quantum searching iterate in its eigenvector basis. Section 2 describes the eigenvectors and eigenvalues of the searching iterate. This analysis immediately show that Grover's algorithm is not very useful when the input state is random, as shown by different methods in [2]. The eigenvalues contain information useful for counting and so in Section 3 we review the techniques for eigenvalue estimation detailed in [10]. Section 4 describes a quantum counting algorithm based on estimating the eigenvalues described in Section 2 with the techniques given in Section 3. The core of this algorithm is, in fact, equivalent to the algorithm in [8], except the analysis is done in a different basis. The analysis in the eigenvector basis is simpler and additional facts become apparent. For example, we can count anywhere from 0 to N solutions and do not need to assume the number of solutions is at most $N/2$ as done in [8]. Analysing the eigenvectors and eigenvalues also provides an alternative analysis of the searching and amplitude amplification algorithms as detailed in [7] and summarised in Section 5. Section 5 also shows how to combine the quantum counting algorithm of Section 4 with exact searching methods to produce an alternative searching algorithm. This algorithm is useful when the number of solutions is not known.

For the rest of the introduction, I will describe a quantum computer and define quantum searching, counting, amplitude amplification, and amplitude estimation in more detail.

1.1. Quantum computers

Consider a register of n-bits and a sequence of logic gates to transform an input to a desired output. Any irreversible gate can be made reversible by adding some fixed number of extra input and output bits, so let us just consider reversible gates (see [1] for information on the history of reversible computation). Since we only observe these n-bits in precisely one of 2^n configurations, we have for centuries assumed this meant

the bits were always in one of these 2^n configurations. Last century we learned that classical physics, which makes such an assumption, is wrong, and we replaced this theory with what is known as *quantum physics*. Such a collection of two-state systems can actually exist in any complex linear combination (or *superposition*) of the 2^n possible observable configurations, provided the coefficients satisfy a certain property. Let us use Dirac's notation and refer to an n-bit string $x = x_1x_2 \ldots x_n$ as $|x\rangle$. The linear combination $\sum_{x\in\{0,1\}^n} \alpha_x|x\rangle$ satisfies $\sum_{x\in\{0,1\}^n} |\alpha_x|^2 = 1$. This restriction occurs for a very good reason: $|\alpha_x|^2$ corresponds to the probability of getting the string $|x\rangle$ if we measure the register. Since we always want to get something, these probabilities better add to 1! The linear combination $\sum_{x\in\{0,1\}^n} \alpha_x|x\rangle$ can also be described in vector notation as $(\alpha_0, \alpha_1, \ldots \alpha_{2^n-1})$. We use the convention that α_j corresponds to the amplitude of $|j\rangle$, where j is represented in binary. Every gate acts linearly on this superposition, so we only need to know the behaviour on a basis of dimension 2^n. We usually use the standard computational basis $\{|x\rangle : x \in \{0,1\}^n\}$, which corresponds to the elementary vectors in the vector space of dimension 2^n generated by the $|x\rangle$ vectors. Further, the restriction that the amplitudes must correspond to probabilities adding up to 1 implies that all the gates are unitary. Consequently, any operator we implement with such gates is unitary.

1.2. Quantum searching and counting

Grover's original quantum searching algorithm [15] takes a function $f : \{0, 1, \ldots, N-1\} \to \{0,1\}$ that has only one solution to $f(x) = 1$ and finds that unique solution using only $O(\sqrt{N})$ evaluations of f. If f is treated as a black box, then $\Omega(\sqrt{N})$ evaluations are in fact necessary [3, 4]. Tighter bounds on the number of evaluations necessary were soon found, the restriction that f has a unique solution was subsequently removed [5], and other algorithms followed that approximately count the number of solutions to $f(x) = 1$ [5, 8, 19].

Let us define the searching and counting problems more explicitly. Consider a function f that maps each element of a set X to either 0 or 1. For example, let X represent the set of the 3^n possible three-colourings of an n-vertex graph $\mathcal{G}$, and let $f(x) = 1$ if and only if the colouring x is a proper colouring of $\mathcal{G}$ (that is, no adjacent vertices are coloured with the same colour). Define X_1 to be the subset of X for which f evaluates to 1 (that is, the set of proper three-colourings of $\mathcal{G}$) and X_0 to be the elements for which f evaluates to 0. Let us define t to be $|X_1|$, the number of elements in X_1.

The *decision* problem associated with f is to decide if there is a proper colouring x, that is, to decide if $|X_1| > 0$. The *generation* or *search* problem is to find a proper colouring x, that is, an element of X_1. The *uniform generation* problem is to generate such an element uniformly at random from the set X_1. A more general problem is to *count* either *exactly* or *approximately* the number of solutions to $f(x) = 1$. To *approximately count* X_1 *with accuracy* ε means to output a number $\tilde{t}$ such that

$$(1-\varepsilon)t \leqslant \tilde{t} \leqslant (1+\varepsilon)t. \tag{1}$$

A *randomised approximation scheme (RAS)* for t is a randomised algorithm that for any real parameter $\varepsilon > 0$ outputs a number $\tilde{t}$ satisfying Eq. (1) with probability $\frac{2}{3}$.[1] A *quantum RAS* is an RAS which uses a quantum computer algorithm.

Grover presented an algorithm for quantum searching [15], which was subsequently generalised [5, 6, 16]. These algorithms do not run in time polynomial in $\log N$, where $|X| = N$, but they do run in time roughly the square root of the running time for the best classical algorithm. By *running time*, we are referring to the number of calls to the *oracle* or *black box* U_f for the function f. This black box for evaluating f reversibly computes $f(x)$ given input $|x\rangle$, usually by mapping $|x\rangle\,|b\rangle$ to $|x\rangle\,|b \oplus f(x)\rangle$, but in this paper we will assume the value of $f(x)$ is simply encoded in the phase by mapping $|x\rangle$ to $(-1)^{f(x)}|x\rangle$. Note that this modified U_f can be realised with a black box which maps $|x\rangle\,|b\rangle$ to $|x\rangle|b \oplus f(x)\rangle$, by setting $|b\rangle$ to $(|0\rangle - |1\rangle)/\sqrt{2}$. In [8] and [19], the iterate in Grover's algorithm, let us call it G, is used to approximately count. The randomised approximation schemes suggested in [5, 8, 19] and herein can be made to run with an expected running time of $O((1/\varepsilon)(\sqrt{N/t}))$ see Lemma 7. We just count the number of calls to U_f since the lower bounds associated with these algorithms are in terms of these calls. It turns out that for all the algorithms discussed here the number of other operations is usually proportional to the number of calls to U_f. The operators A and A^{-1} we discuss later are typically Hadamard transforms or some other transformations which can be efficiently implemented, and the operator U_0 can also be implemented efficiently. Since the algorithms in this paper use only one application of A, A^{-1}, and U_0 for every application of U_f, this measure of running time is indeed representative of the running time of these algorithms in terms of all the elementary operations necessary. Each G makes one call to U_f, so the number of repetitions of G corresponds to the number of calls to U_f. In the next section we take a closer look at the operator G and its properties.

2. The Grover iterate and its properties

The quantum searching algorithm [15, 5] prepares the state

$$\frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle,$$

where we identify the n-bit strings with the integers from 0 to $2^n - 1$. It then iterates the operator

$$G = -AU_0A^{-1}U_f.$$

[1] The number $\frac{2}{3}$ can be replaced by any value, say $1 - \delta$, that exceeds $\frac{1}{2}$ by a constant. Given a particular RAS (see e.g. [22]), we can apply a *bootstrapping scheme* that applies the given RAS a number of times linear in $\log(1/\delta)$ and outputs the median to produce an ε-approximation with probability $1 - \delta$. See Lemma 6.1 of [17].

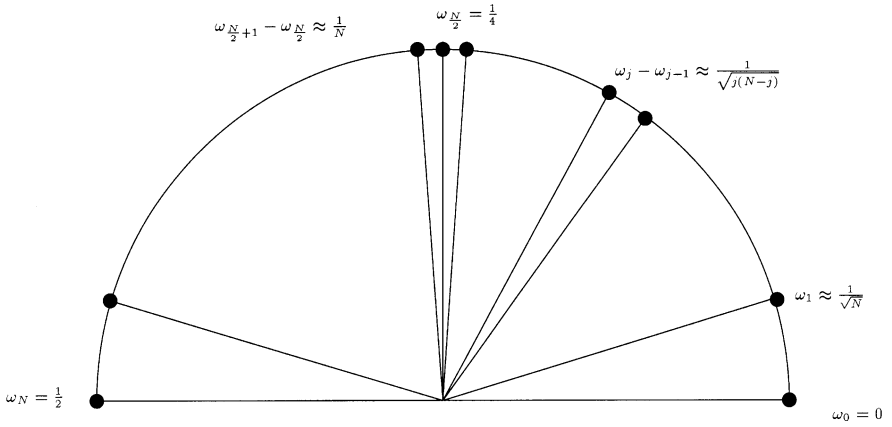

Fig. 1. The eigenvalue of G on $|\Psi_+\rangle$ when there are t solutions is $e^{2\pi i\omega_t}$. This diagram illustrates the distribution of ω_t depending on the number of solutions t. The important point is that the ω_t values get much closer together and harder to differentiate as t gets close to $N/2$. Distinguishing a function f with t solutions requires a more precise estimate of ω_t as t gets closer to $N/2$.

Here A is any operator which maps $|0\rangle$ to $(1/\sqrt{N})\sum_{x=0}^{N-1}|x\rangle$, U_0 maps $|0\rangle$ to $-|0\rangle$ and leaves the remaining $|x\rangle$ alone, and U_f maps $|x\rangle$ to $(-1)^{f(x)}|x\rangle$.

Recall that $t=|X_1|$, the number of solutions to $f(x)=1$, and $|X_0|=N-t$. When $t=0$ or N, $A|0\rangle$ is an eigenvector of G with eigenvalue $e^{\pi it/N}$. For $0<t<N$ define

$$|X_1\rangle = \frac{1}{\sqrt{t}}\sum_{x\in X_1}|x\rangle, \tag{2}$$

$$|X_0\rangle = \frac{1}{\sqrt{N-t}}\sum_{x\in X_0}|x\rangle, \tag{3}$$

$$|\Psi_+\rangle = \frac{1}{\sqrt{2}}|X_1\rangle + \frac{\mathrm{i}}{\sqrt{2}}|X_0\rangle, \tag{4}$$

$$|\Psi_-\rangle = \frac{1}{\sqrt{2}}|X_1\rangle - \frac{\mathrm{i}}{\sqrt{2}}|X_0\rangle. \tag{5}$$

Since A is unitary and maps $|0\rangle$ to $|X_1\rangle$ with amplitude $\sqrt{t/N}$, then A^{-1} must map $|X_1\rangle$ to $|0\rangle$ with amplitude $\sqrt{t/N}$. One consequence is that $AU_0A^{-1}|X_1\rangle = |X_1\rangle - 2\sqrt{t/N}A|0\rangle$. Using these and similar facts we can verify that $|\Psi_+\rangle$ and $|\Psi_-\rangle$ are eigenvectors of G with respective eigenvalues $e^{2\pi i\omega}$ and $e^{-2\pi i\omega}$, $0<\omega<\frac{1}{2}$, with $\cos(2\pi\omega)=1-2t/N$, and $\sin(2\pi\omega)=2\sqrt{t(N-t)}/N$.

Define $\omega_0=0$, $\omega_N=\frac{1}{2}$, and for t strictly in between 0 and N define ω_t so that the eigenvalue of the eigenvector $|\Psi_+\rangle$ of $G=-AU_0A^{-1}U_f$ is $e^{2\pi i\omega_t}$ and $0\leqslant\omega_t\leqslant\frac{1}{2}$ (see Fig. 1). Consequently,

$$\cos(2\pi\omega_t) = 1-\frac{2t}{N}, \quad \sin(2\pi\omega_t) = 2\frac{\sqrt{t(N-t)}}{N} \quad \text{and} \quad \sin(\pi\omega_t) = \sqrt{\frac{t}{N}}. \tag{6}$$

It follows that

$$2\pi\omega_t = \arccos\left(1 - \frac{2t}{N}\right) = 2\sqrt{\frac{t}{N}} + \mathrm{O}\left(\left(\frac{t}{N}\right)^{3/2}\right). \tag{7}$$

Also note that, for t strictly between 0 and N,

$$\frac{1}{\sqrt{2}}|\Psi_+\rangle + \frac{1}{\sqrt{2}}|\Psi_-\rangle = |X_1\rangle,$$

which we seek since measuring $|X_1\rangle$ solves the uniform generation problem for f. We start with the state $A|0\rangle = \sqrt{t/N}|X_1\rangle + \sqrt{(N-t)/N}|X_0\rangle = \sin(\pi\omega_t)|X_1\rangle + \cos(\pi\omega_t)\,|X_0\rangle$, which is expressed in the eigenvector basis as

$$\frac{-\mathrm{i}e^{\pi\mathrm{i}\omega_t}}{\sqrt{2}}|\Psi_+\rangle + \frac{\mathrm{i}e^{-\pi\mathrm{i}\omega_t}}{\sqrt{2}}|\Psi_-\rangle.$$

To determine or estimate $t = |X_1|$, we will estimate the phase $2\pi\omega_t$. How accurately should we estimate ω_t to determine t in the worst case? There are $N+1$ such ω_t for $t = 0, 1, \ldots, N$, all between 0 and $\frac{1}{2}$, so by the Pigeon Hole Principle at least 2 of them are at most distance $1/2N$ apart and our phase estimation will have to be quite accurate to distinguish all of them; we would require on the order of N applications of G.

More precisely, we have the following lemmas by the Mean Value Theorem and the fact that the derivative of ω_t as a function of t is $1/(2\pi\sqrt{t(N-t)})$.

Lemma 1. *For any integer t satisfying $0 \leqslant t < N/2$,*

$$\frac{1}{\sqrt{(t+1)(N-t-1)}} \leqslant 2\pi|\omega_{t+1} - \omega_t|$$

and for $0 < t \leqslant N/2$,

$$2\pi|\omega_{t+1} - \omega_t| \leqslant \frac{1}{\sqrt{t(N-t)}}.$$

Lemma 2. *For any integer t satisfying $0 \leqslant t \leqslant N/4$,*

$$2\pi|\omega_{2t} - \omega_t| \leqslant \sqrt{\frac{t}{N}}.$$

It is worth remembering that there are of course many other eigenvectors. If $0 < t < N$, then in addition to the eigenvectors $|\Psi_+\rangle$ and $|\Psi_-\rangle$ there are $N-2$ other eigenvectors. Exactly $N-t-1$ of them, spanned only by elements of X_0, have eigenvalue -1 and $t-1$ of them, spanned only by elements of X_1, have eigenvalue 1. It is easy to find a spanning set of these eigenvectors. One interesting use of this fact is to study the effect of applying the quantum searching algorithms with arbitrary input states. The optimal number of applications of G before measuring was studied in [2] (by different methods). Applying G has no effect on the eigenvectors with eigenvalue 1, and flips

the sign in front of the eigenvectors with eigenvalue -1. Consequently, on states with zero amplitude for the eigenvectors $|\Psi_+\rangle$ and $|\Psi_-\rangle$, G is equivalent to a simple $-U_f$! So unless the amplitudes of $|\Psi_+\rangle$ and $|\Psi_-\rangle$ in the initial state are significant, which is unlikely if we start in a 'random' state, Grover's algorithm will be of no help in searching.

In the next section we describe algorithms for estimating phases corresponding to eigenvalues of unitary operators.

3. Quantum phase estimation

Here we will review the relationship, as pointed out in [10], between the quantum Fourier transform and the estimation of phases. For any integer $M>1$, the quantum Fourier transform, F_M, for each integer z from 0 to $M-1$, maps $|z\rangle$ to

$$\frac{1}{\sqrt{M}}\sum_{y=0}^{M-1} e^{2\pi i(z/M)y}|y\rangle. \tag{8}$$

Clearly, for any $\omega=z/M$, where z is an integer between 0 and $M-1$, the inverse Fourier transform, F_M^{-1}, maps the state in Eq. (8) to $|z\rangle$.

Given any real number ω satisfying $0\leqslant\omega<1$ encoded in the phases of the superposition

$$\frac{1}{\sqrt{M}}\sum_{y=0}^{M-1} e^{2\pi i\omega y}|y\rangle, \tag{9}$$

applying the inverse quantum Fourier transform, F_M^{-1}, will map this superposition to a superposition

$$\frac{1}{\sqrt{M}}\sum_{z=0}^{M-1} \alpha_z|z\rangle, \tag{10}$$

which we will denote by $|\tilde{\omega}\rangle$. The amplitudes are concentrated near the values of z for which z/M is a good estimate of ω. More precisely, we have Lemmas 3 and 4 (see [10]). Let $d(a,b)$ denote the distance between a and b modulo 1 (see Fig. 2).

Lemma 3. *When observing the state in Eq.* (10), *the probability of obtaining* $|y\rangle$ *such that* $d(\omega, y/M)\leqslant 1/(2M)$ *is at least* $4/\pi^2$. *This fraction* y/M *corresponds to the best estimate of* ω *as a fraction of* M.

We can replace $4/\pi^2$ with any $1-\delta$, $0<\delta<1$, by estimating ω using $M'=\lceil 1/(2\delta)+1/2\rceil M$ instead of M, and then rounding off the estimate to a fraction of M.

Lemma 4. *For any positive integer* $k<M$, *when observing the state in Eq.* (10), *the probability of obtaining a state* $|y\rangle$ *with* $d(\omega, y/M)\leqslant k/(2M)$ *is at least* $1-1/(2k-1)$.

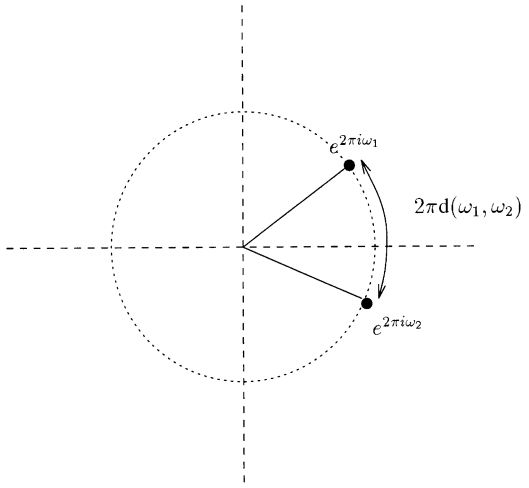

Fig. 2. We define the distance between the real numbers ω_1 and ω_2, $d(\omega_1, \omega_2)$, to be the smallest real number d between 0 and $\frac{1}{2}$ such that $\mathrm{e}^{2\pi \mathrm{i}(\omega_1-\omega_2)}$ equals one of $\mathrm{e}^{2\pi \mathrm{i}d}$ or $\mathrm{e}^{-2\pi \mathrm{i}d}$. In other words, it is length of the shortest path (scaled by $1/2\pi$) along the unit circle from $\mathrm{e}^{2\pi \mathrm{i}\omega_1}$ to $\mathrm{e}^{2\pi \mathrm{i}\omega_2}$.

Thus given an operator G with eigenvector $|\Psi\rangle$ and eigenvalue $\mathrm{e}^{2\pi \mathrm{i}\omega}$, we can estimate ω as follows. Prepare the state

$$\frac{1}{\sqrt{M}} \sum_{y=0}^{M-1} |y\rangle |\Psi\rangle \tag{11}$$

and apply G to $|\Psi\rangle$ y times when the first register is in state $|y\rangle$. This can be done in superposition, that is, without observing $|y\rangle$, by using controlled-G operations. A controlled-G is like a G but with an additional control qubit. If the control qubit is in state $|1\rangle$ then G is applied, and if the control qubit is in state $|0\rangle$ then G is not applied. We represent $|y\rangle$ in binary as $|y_{n-1}\rangle|y_{n-2}\rangle \ldots |y_0\rangle$, where $y = 2^{n-1}y_{n-1} + 2^{n-2}y_{n-2} + \cdots + y_0$, $y_i \in \{0, 1\}$. We use the qubit $|y_j\rangle$ as a control bit to implement 2^j controlled-G operations, which is equivalent to a controlled-G^{2j}. This procedure (see Fig. 3) creates the state

$$\frac{1}{\sqrt{M}} \sum_{y=0}^{M-1} \mathrm{e}^{2\pi \mathrm{i}\omega y} |y\rangle |\Psi\rangle. \tag{12}$$

Applying F_M^{-1} to the first register gives the state $|\tilde{\omega}\rangle|\Psi\rangle$, and has the property that when we observe the first register we get an estimate $\tilde{\omega}$ of ω.

Suppose we are just given the operator U_f as a black box and the state $|\Psi\rangle$ such that $G|\Psi\rangle = \mathrm{e}^{2\pi \mathrm{i}\omega}|\Psi\rangle$, and we only interact with $|\Psi\rangle$ by applying G to it. We have the following two lemmas.

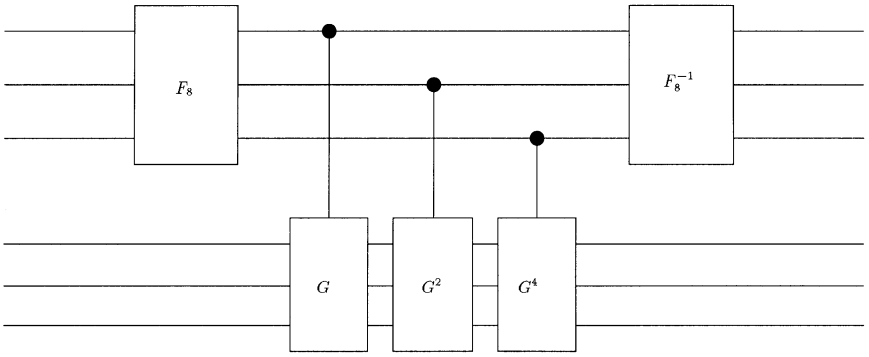

Fig. 3. This diagram illustrated a small network for estimating an eigenvalue of G.

Lemma 5. *For any $\varepsilon>0$, we can obtain an estimate $\tilde{\omega}$ of ω so that $d(\omega,\tilde{\omega})<\varepsilon$ with probability $\geqslant\frac{2}{3}$ with $O(1/\varepsilon)$ applications of G. We can also replace $\frac{2}{3}$ by any constant $p<1$.*

Proof. The sufficiency of $M=\lfloor 1/\varepsilon\rfloor+1$ applications follows from Lemma 4 by setting $k=2$. For any other probability $p<1$, we set $k=\lfloor 1/(2-2p)\rfloor+1$ and $M=k\lfloor 1/(2\varepsilon)\rfloor+1$. □

Lemma 6. *For ε between $1/N$ and $1/\sqrt{N}$, to obtain an estimate $\tilde{\omega}$ of ω so that $d(\omega,\tilde{\omega})<\varepsilon$ with probability $\geqslant\frac{2}{3}$, requires $\Omega(1/\varepsilon)$ applications of G.*

Proof. In [4] it is shown that to decide, with error probability at most $\frac{1}{3}$, if a Boolean function f has fewer than M solutions to $f(x)=1$, for $0<M\leqslant N/2$, requires $\Omega(\sqrt{NM})$ calls to U_f. Lemma 1 tells us that by estimating ω_t within $1/(2\sqrt{M(N-M)})$ with error at most $\frac{1}{3}$ will solve this problem for us. The lower bound now follows by setting $M=\lceil 1/(2N\varepsilon^2)\rceil$. □

4. Quantum counting

We are now ready to combine the facts about the eigenvectors and eigenvalues of G in Section 2 with the techniques in Section 3 to approximately count, t, the number of solutions to $f(x)=1$. The parameter M represents the number of times we will iterate G and thus corresponds to the running time of the algorithm in terms of evaluations of U_f. It is chosen depending on the quality we desire for the estimate. Start with the state

$$\frac{1}{\sqrt{M}}\sum_{y=0}^{M-1}|y\rangle A|0\rangle,$$

which, assuming $0<t<N$, is equal to

$$\frac{-\mathrm{i}e^{\pi i\omega_t}}{\sqrt{2}}\sum_{y=0}^{M-1}|y\rangle|\Psi_+\rangle+\frac{\mathrm{i}e^{\pi i\omega_t}}{\sqrt{2}}\sum_{y=0}^{M-1}|y\rangle|\Psi_-\rangle.$$

Then apply G to the second register y times when the first register is in state $|y\rangle$ to produce

$$\frac{-\mathrm{i}e^{\pi i\omega_t}}{\sqrt{2}}\sum_{y=0}^{M-1}e^{2\pi i\omega_t y}|y\rangle|\Psi_+\rangle+\frac{\mathrm{i}e^{-\pi i\omega_t}}{\sqrt{2}}\sum_{y=0}^{M-1}e^{-2\pi i\omega_t y}|x\rangle|\Psi_-\rangle.$$

Lastly apply F_M^{-1} to the first register to produce

$$\frac{-\mathrm{i}e^{\pi i\omega_t}}{\sqrt{2}}|\widetilde{\omega_t}\rangle|\Psi_+\rangle+\frac{\mathrm{i}e^{-\pi i\omega_t}}{\sqrt{2}}|\widetilde{-\omega_t}\rangle|\Psi_-\rangle.$$

If $t=0$ or N, the same operations would produce $|\widetilde{\omega_t}\rangle A|0\rangle$.

Recall the definition of $|\widetilde{\omega_t}\rangle$ from Section 3. It is a superposition whose amplitudes are concentrated near values of y such that y/M is a close estimate of ω_t. Measuring the first register will output (each with probability $\frac{1}{2}$) either an estimate of ω_t, or of $1-\omega_t$ (if $t=0$ or N, we just estimate ω_t, which is equivalent to $1-\omega_t$ modulo 1). When we measure an integer y between 0 and $M/2$, we will estimate ω_t with the number $\widetilde{\omega_t}=y/M$. If we measure an integer y between $M/2$ and M we will estimate ω_t with the number $1-y/M$. It is easy to see that this protocol will produce an estimate of ω_t that is no worse than if we only measured $|\widetilde{\omega_t}\rangle|\Psi_+\rangle$ (that is, the probability of getting an error greater than ε does not increase for any $\varepsilon>0$).

So let us assume that $\widetilde{\omega_t}=y/M$ is our estimate of ω_t. Define $\varepsilon=\omega_t-\widetilde{\omega_t}$. We know that

$$\cos(2\pi y/M)=\cos(2\pi\omega_t)\cos(-2\pi\varepsilon)-\sin(2\pi\omega_t)\sin(-2\pi\varepsilon). \tag{13}$$

With $O(M)$ applications of G we can obtain an estimate such that with probability at least $\frac{2}{3}$ we have $|2\pi\varepsilon|\leqslant 1/M$ (see Section 3), and so $|\cos(2\pi\varepsilon)-1|\leqslant 1/2M^2$ and $|\sin(2\pi\varepsilon)|\leqslant 1/M$. Using Eq. (6) we get an estimate for t:

$$\tilde{t}=N\frac{(1-\cos(2\pi y/M))}{2}=N\left(\sin^2\left(\frac{\pi y}{M}\right)\right). \tag{14}$$

By Eq. (13) and the above bounds on $\cos(2\pi\varepsilon)$ and $\sin(2\pi\varepsilon)$, we have that with probability at least $\frac{2}{3}$,

$$|\tilde{t}-t|\leqslant\frac{|N-2t|}{4M^2}+\frac{\sqrt{t(N-t)}}{M}. \tag{15}$$

By Lemma 4, with probability $1-O(1/k)$ we have $|2\pi\varepsilon|\leqslant k/M$ and similarly

$$|\tilde{t}-t|\leqslant\frac{|N-2t|k^2}{4M^2}+\frac{k\sqrt{t(N-t)}}{M}. \tag{16}$$

Some corollaries (similar to ones in [5, 8, 19]) are the following.

Corollary 1. *If* $M = \lceil c\sqrt{N} \rceil$, *then with probability at least* $\frac{2}{3}$ *we will have*

$$|\tilde{t} - t| \leqslant \frac{1}{4c^2} + \frac{\sqrt{\min(t, N-t)}}{c} \in \mathrm{O}\left(\frac{\sqrt{\min(t, N-t)}}{c}\right).$$

Corollary 2. *If* $M = \lceil c\sqrt{N/(t+1)} \rceil$, *then with probability at least* $\frac{2}{3}$ *we will have*

$$(1-\varepsilon)t \leqslant \tilde{t} \leqslant (1+\varepsilon)t,$$

where $\varepsilon = 1/(4c^2\sqrt{t+1}) + 1/c \in \mathrm{O}(1/c)$.

With a simple eigenvalue estimation protocol, this author, together with the authors of [8], improved the running time $\mathrm{O}((1/\varepsilon + \log\log(N))\sqrt{N/(t+1)})$ of the approximate counting algorithms in [8] and [19] (see [7]).

Lemma 7. *There is a quantum RAS for the number of solutions,* t, *to* $f(x) = 1$, $0 \leqslant x < N$, *with running time* $\mathrm{O}((1/\varepsilon)\sqrt{N/(t+1)})$.

Corollary 1 gives us to the bound on t that we need to carry out exact counting (combining Lemmas 4 and 1).

Corollary 3. *Given* $G = -AU_0A^{-1}U_f$, *where* f *has* t *solutions to* $f(x) = 1$, *we can distinguish* ω_t *and correctly determine* t *with probability at least* $\frac{2}{3}$, *and the expected number of applications of* G *is only* $\Theta(\sqrt{(t+1)(N-t+1)})$.

Proof. By the symmetry between ω_t and ω_{N-t}, we can assume $0 \leqslant t \leqslant N/2$. Using $\Theta(\sqrt{N})$ applications of G, estimate t with $\tilde{t}$ so that $|t - \tilde{t}| < 1 + \sqrt{t}$ with high probability, say $\frac{4}{5}$ (use Corollary 1). By Eq. (16) and Lemma 4 we can assume $|t - \tilde{t}| \leqslant k^2 + k\sqrt{t}$ with probability $1 - \mathrm{O}(1/k)$, for $k > 0$. Repeat this procedure five times and take the median of the $\tilde{t}$ values. Use $\mathrm{O}(\sqrt{(\tilde{t}+1)(N-\tilde{t}+1)})$ iterations of G to estimate t again with $\bar{t}$. Choose the constant in $\mathrm{O}(\sqrt{(\tilde{t}+1)(N-\tilde{t}+1)})$ so that if $|t - \tilde{t}| < 1 + \sqrt{t}$, then $\bar{t} = t$ with probability $\frac{5}{6}$. For $k > \sqrt{t}/3$, the median $\tilde{t}$ of the 5 estimates for t will satisfy $t - 2k^2 < \tilde{t} < t + 2k^2$ with probability $1 - \mathrm{O}(1/k^3)$ (at least 3 of the 5 estimates must satisfy $|t - \tilde{t}| > k^2 + k\sqrt{t}$ for this not to happen). If $k \leqslant \sqrt{t}/3$ then $|t - \tilde{t}| < 4t/9$ and the running time is still $\Theta(\sqrt{(t+1)(N-t+1)})$ iterations of G. The expected running time when $k > \sqrt{t}/3$ also converges to $\Theta(\sqrt{(t+1)(N-t+1)})$ iterations of G since $\sqrt{(t+2k^2+1)(N-t-2k^2+1)}) \leqslant \sqrt{(2k^2+1)(t+1)(N-t+1)})$ $\in \mathrm{O}(k\sqrt{(t+1)(N-t+1)})$ and $\sum_{k=1}^{\infty} 1/k^2 = \pi^2/6$. Thus the expected running time of the whole algorithm is $\Theta(\sqrt{(t+1)(N-t+1)})$ iterations of G. The probability that $\bar{t} = t$ is at least $\frac{2}{3}$. □

In Section 6 we point out how this algorithm is a special case of *amplitude estimation*.

5. Quantum searching

The quantum searching algorithm [15, 5] can be succinctly analysed as follows. Start in the state

$$A|0\rangle = \frac{-\mathrm{i}}{\sqrt{2}}\mathrm{e}^{\pi\mathrm{i}\omega_t}|\Psi_+\rangle + \frac{\mathrm{i}}{\sqrt{2}}\mathrm{e}^{-\pi\mathrm{i}\omega_t}|\Psi_-\rangle,$$

apply G to this state k times to produce

$$\frac{-\mathrm{i}}{\sqrt{2}}\mathrm{e}^{\pi\mathrm{i}(2k+1)\omega_t}|\Psi_+\rangle + \frac{\mathrm{i}}{\sqrt{2}}\mathrm{e}^{-\pi\mathrm{i}(2k+1)\omega_t}|\Psi_-\rangle, \tag{17}$$

and measure. The number of repetitions k will correspond to the running time of the algorithm. Since we want to observe $|X_1\rangle = \frac{1}{\sqrt{2}}|\Psi_+\rangle + \frac{1}{\sqrt{2}}|\Psi_-\rangle$, we need to align the phases so that the relative phase between the two eigenvectors is close to 0. When t is known, obtaining a relative phase close to 0 in Eq. (17) is easy. We just pick k so that $-\mathrm{e}^{\pi\mathrm{i}(2k+1)\omega_t} = \mathrm{e}^{-\pi\mathrm{i}(2k+1)\omega_t}$, or $(4k+2)\omega_t$ is an odd integer. When t is small this occurs when k is roughly $(\pi/4)\sqrt{N/t}$. Also consider, as done in [5], the case that $t = N/4$. We have $\omega_t = \frac{1}{6}$ so we get $|X_1\rangle$ with exactly $k = 1$ iterations of G. As an easy special case of a result in [2], we note that if we start off with the state

$$c\mathrm{e}^{\pi\mathrm{i}\theta}|\Psi_+\rangle + d\mathrm{e}^{-\pi\mathrm{i}\theta}|\Psi_-\rangle,$$

where c and d are positive reals, then to maximise the amplitude of the states $|x\rangle$ with $f(x) = 1$ we should apply G to the starting state k times where $2(2k\omega_t + \theta)$ is close to an odd integer.

When t is not known, it is not as simple. One idea is to estimate ω_t using the techniques of the previous section, and then pick the number of repetitions k.

A different approach is given in [5] and a similar quantum version is given later in [19]. Both approaches search over an interval of increasing size, finding a solution at each observation with probability approaching $\frac{1}{2}$. To control the expected running time in [5] the authors make sure the sizes of the intervals increase by a factor less than 2 at each iteration, while in [19] a bootstrapping method is applied together with incrementing the interval size appropriately. At the end of the next section we describe a method for which the probability of success tends to 1 as we increase the interval size M. The searching algorithm described in this section effectively takes the $\sqrt{N/t}$ probability amplitude of success we have with a uniform superposition starting state and amplifies this probability to 1. However, there is no reason to restrict ourselves to such a naive starting state. We discuss this further in the next section. With the algorithm we have just described there is usually a small chance of failing to generate a solution, even if we know how many there are. The next section also reviews how to make the probability of success equal to 1 when we know the probability of success.

6. Amplitude amplification and estimation

In [6, 16] we see that we can, in fact, replace A with any transformation which maps $|0\rangle$ to $\sqrt{a}|X_1\rangle+\sqrt{b}|X_0\rangle$, where $|X_1\rangle$ is *any* superposition (of norm 1) of basis states $|x\rangle$ satisfying $f(x)=1$, $|X_0\rangle$ is *any* superposition (of norm 1) of basis states $|x\rangle$ satisfying $f(x)=0$, and a and b are positive reals satisfying $a+b=1$.

The eigenvectors of $G=-AU_0A^{-1}U_f$ have the same form as in Eqs. (4) and (5):

$$|\Psi_+\rangle=\frac{1}{\sqrt{2}}(|X_1\rangle+\mathrm{i}|X_0\rangle),$$

$$|\Psi_-\rangle=\frac{1}{\sqrt{2}}(|X_1\rangle-\mathrm{i}|X_0\rangle).$$

They have corresponding eigenvalues $\mathrm{e}^{2\pi\mathrm{i}\omega_a}$ and $\mathrm{e}^{-2\pi\mathrm{i}\omega_a}$ where $0\leqslant\omega_a\leqslant\frac{1}{2}$, $\cos(2\pi\omega_a)=1-2a$, $\sin(2\pi\omega_a)=2\sqrt{a(1-a)}$, $\sin(\pi\omega_a)=\sqrt{a}$, and

$$A|0\rangle=-\frac{\mathrm{i}}{\sqrt{2}}\mathrm{e}^{2\pi\mathrm{i}\omega_a}|\Psi_+\rangle+\frac{\mathrm{i}}{\sqrt{2}}\mathrm{e}^{-2\pi\mathrm{i}\omega_a}|\Psi_-\rangle$$

(ω_t from the previous sections corresponds to $\omega_{t/N}$ here).

We can thus apply the same searching technique described in Section 5 to amplify the amplitude of $|X_1\rangle$ to 1, only requiring an expected $\mathrm{O}(\sqrt{1/a})$ applications of G for $a>0$. We can use the same techniques of Section 4 to approximate a (*amplitude estimation*). That is, we estimate an eigenvalue of G, say $\mathrm{e}^{2\pi\mathrm{i}\omega_a}$, with $\widetilde{\omega_a}$, and then estimate a with $\sin^2(\pi\widetilde{\omega_a})$. When we know a we know exactly how many applications of G we should use to search for elements of X_1. We can also alter G slightly so that the ideal number, M, of applications is an integer, making the search exact. This was first done for $M=1$ in [6], and later in [11], by altering the phase shifts in U_0 and U_f. This method is also used for any $M>0$ in [8] and [7]. Another simple method is to modify A and f so that a is slightly smaller but can be amplified to 1 with an integer number of iterations of G (see [8, 19] or [7]). In the next section we present a scheme for which the probability of observing a solution tends to 1 as the interval size, M, tends to infinity. Thus at each iteration we can increase M by any constant factor and still control the expected running time.

6.1. Amplifying an unknown amplitude

Note that when we estimate an amplitude and produce the state

$$\frac{-\mathrm{i}\mathrm{e}^{\pi\mathrm{i}\omega_a}}{\sqrt{2}}|\widetilde{\omega_a}\rangle|\Psi_+\rangle+\frac{\mathrm{i}\mathrm{e}^{-\pi\mathrm{i}\omega_a}}{\sqrt{2}}|\widetilde{-\omega_a}\rangle|\Psi_-\rangle, \tag{18}$$

we could also measure the second register and test the answer, $|x\rangle$, to see if $f(x)=1$. Denote by S the operator, as described in Section 4, which maps two registers of zeros, $|0\rangle|0\rangle$, to the state in Eq. (18).

Note that as $|\widetilde{\omega_a}\rangle$ and $|\widetilde{-\omega_a}\rangle$ become better estimates of ω_a and $-\omega_a$, then, unless $a=0$ or 1,

$$|\langle\widetilde{\omega_a}|\widetilde{-\omega_a}\rangle| = \frac{1}{M}\left|\sum_{x=0}^{M-1} e^{4\pi i\omega_a}\right| = \frac{1}{M}\left|\frac{\sin(2M\pi\omega_a)}{\sin(2\pi\omega_a)}\right| \leqslant \frac{1}{4M\omega_a}.$$

Measuring either $|\Psi_+\rangle$ or $|\Psi_-\rangle$ will reveal an element of X_1 with probability $\frac{1}{2}$. Thus, when we measure the second register of state (18) we get an element of X_1 with probability at least $\frac{1}{2} - O(1/(M\sqrt{a}))$ as $M\sqrt{a} \to \infty$.

To keep matters simple, let us assume that $0 < a \leqslant \frac{1}{2}$, and let us modify the algorithm slightly to produce an algorithm S that has probability of success $\frac{1}{4} - O(1/(M\sqrt{a}))$ as $M\sqrt{a} \to \infty$ (it is easy to *reduce* the probability of success!). For any algorithm S which outputs a solution with probability $\sin^2(\pi\omega) = \frac{1}{4} - \varepsilon$, for some small ε, observing after one application of the searching iterate $G = -SU_0S^{-1}U_f$ to the state $S|0\rangle$ will produce a solution with probability $\sin^2(3\pi\omega)$. By a simple trigonometric identity this probability equals $\sin^2(\pi\omega)(3 - 4\sin^2(\pi\omega))^2 = 1 - 12\varepsilon^2 - 16\varepsilon^3$.

Thus instead of observing after each application of S and carefully incrementing the interval size in case of failure (as done classically in [5, 8] and quantumly in [19]), we can use this operator S and its inverse S^{-1} as a subroutine in a slightly larger algorithm. Namely, we can apply $SU_0S^{-1}U_fS$ to the state $|0\rangle$ to get an algorithm which works with probability $1 - O(1/(M^2a))$ as $M\sqrt{a} \to \infty$. In case of failure, we can increment our interval size by any constant multiple greater than 1. The expected running time of this algorithm is $O(1/\sqrt{a})$ applications of G for $a > 0$. For $a = 0$ it would run forever.

Acknowledgements

Many thanks to Artur Ekert for many helpful discussions and for emphasising to me the relationship between quantum computing and interferometry. Thanks also to Gilles Brassard, Richard Cleve, Peter Høyer, Alain Tapp, and Ronald deWolf for helpful discussions and comments. I am grateful to C.E.S.G. for their financial support.

References

[1] C.H. Bennett, Notes on the history of reversible computation, IBM J. Res. Dev. 32 (1988) 16–23.

[2] D. Biron, O. Biham, E. Biham, M. Grassl, D.A. Lidar, Generalized Grover Search Algorithm for Arbitrary Initial Amplitude Distribution, Proc. 1st NASA Internat. Conf. on Quantum Computing and Quantum Communications, Palm Springs, USA, Lecture Notes in Computer Science, 1509, 1999, pp. 140–147.

[3] C.H. Bennett, E. Bernstein, G. Brassard, U.V. Vazirani, Strengths and weaknesses of quantum computing, SIAM J. Comput. 26 (1997) 1510–1523.

[4] R. Beals, H. Buhrman, R. Cleve, M. Mosca, R. de Wolf, Quantum lower bounds by polynomials, Proc. 39th Ann. Symp. on the Foundations Comput. Sci., 1998, pp. 352–361.

[5] M. Boyer, G. Brassard, P. Høyer, A. Tapp, Tight Bounds on Quantum Searching, Proc. 4th Work. on Physics and Computation, 1996, pp. 36–43. Journal version appeared in Fortschr. Phys. 46 (1998) 493–505.
[6] G. Brassard, P. Høyer, An exact quantum polynomial-time algorithm for Simon's problem, Proc. of the 5th Israeli Symp. on Theory of Computing and Systems, 1997, IEEE Computer Society Press, Silver Spring, MD, pp. 12–23.
[7] G. Brassard, P. Høyer, M. Mosca, A. Tapp, Quantum amplitude amplification and estimation, preprint.
[8] G. Brassard, P. Høyer, A. Tapp, Quantum counting, Proc. 25th Internat. Coll. on Automata, Languages and Programming, Lecture Notes in Computer Science, vol. 1443, 1998, pp. 820–831.
[9] E. Bernstein, U. Vazirani, Quantum Complexity Theory, Proc. 25th Annual ACM Symposium on the Theory of Computing, 1993, ACM Press, New York, pp. 11–20. Journal version appeared in SIAM J. Comput. 26 (1997) 1474–1483.
[10] R. Cleve, A. Ekert, C. Macchiavello, M. Mosca, Quantum algorithms revisited, Proc. Roy. Soc. London A 454 (1998) 339–354.
[11] D.P. Chi, J. Kim, Quantum database searching by a single query, Proc. of the 1st NASA Internat. Conf. on Quantum Computing and Quantum Communication, 148–151.
[12] D. Deutsch, Quantum-theory, the Church-Turing principle and the universal quantum computer, Proc. Roy. Soc. London A 400 (1985) 97–117.
[13] D. Deutsch, R. Jozsa, Rapid solution of problems by quantum computation, Proc. Roy. Soc. London A, 439 (1992) 553–558.
[14] R.P. Feynman, Simulating Physics with computers, Internat. J. Theor. Phys. 21 (1982) 467–488.
[15] L.K. Grover, A fast quantum mechanical algorithm for database search, Proc. 28th Ann. ACM Symp. on the Theory of Computing, 1996, ACM Press, New York, pp. 212–219. Journal version, Quantum Mechanics helps in searching for a needle in a haystack, appeared in Physical Review Letters, 79 (1997) 325–328.
[16] L.K. Grover, A framework for fast quantum mechanical algorithms, Proc. 30th Ann. ACM Symp. on the Theory of Computing, 1998, ACM Press, New York, pp. 53–62.
[17] M.R. Jerrum, L.G. Valiant, V.V. Vazirani, Random generation of combinatorial structures from a uniform distribution, Theoret. Comput. Sci. 43 (1986) 169–188.
[18] A.Y. Kitaev, Quantum measurements and the Abelian Stabilizer Problem, 1995, http://xxx.lanl.gov/abs/quant-ph/9511026.
[19] M. Mosca, Quantum Searching, Counting and Amplitude Amplification by Eigenvector Analysis, in: R. Freivalds (Ed.), Proc. of Randomized Algorithms, Workshop of Mathematical Foundations of Computer Science, Brno, Czech Republic, 1998, pp. 90–100.
[20] M. Mosca, A. Ekert, The hidden subgroup problem and eigenvalue estimation on a quantum computer, Proc. of the 1st NASA Int. Conf. on Quantum Computing and Quantum Communication, Lecture Notes in Computer Science, 174–188.
[21] A.J. Menezes, P.C. van Oorschot, S.A. Vanstone, Handbook of Applied Cryptography, CRC Press, London, 1996.
[22] R. Motwani, P. Raghavan, Randomized Algorithms, Cambridge University Press, Cambridge, 1995.
[23] P.W. Shor, Algorithms for quantum computation: discrete logarithms and factoring, Proc. 35th Ann. Symp. on Foundations of Comp. Sci., 1994, pp. 124–134. Journal version, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, SIAM J. Comput. 26 (1997) 1484–1509.
[24] D. Simon, On the Power of Quantum Computation, Proc. 35th Ann. Symp. on Foundations Comput. Sci. (1994) 116–123. Journal version appeared in SIAM J. Comput. 26 (1997) 1474–1483.